CS 295A/395D: Artificial Intelligence

Uncertain States

Prof. Emma Tosch

16 March 2022

The University of Vermont

Agenda

- Student hours – extended starting next week (will announce online)
- Recap: agents + search
- What is uncertainty in state
- Representing uncertainty in state with Bayes nets
- Recent research

- But first…

Dr Valeria dePaiva and 4 others liked

Melanie Mitchell
@MelMitchell1

Amidst all the recent Twitter talk on the role of symbols vs. deep learning in AI, I came across a very interesting article by Allen Newell, published in 1982, entitled "Intellectual Issues in the History of Artificial Intelligence".

apps.dtic.mil/sti/pdfs/ADA12...

🧵 (1/9)

7:30 PM · Mar 14, 2022 · Twitter Web App

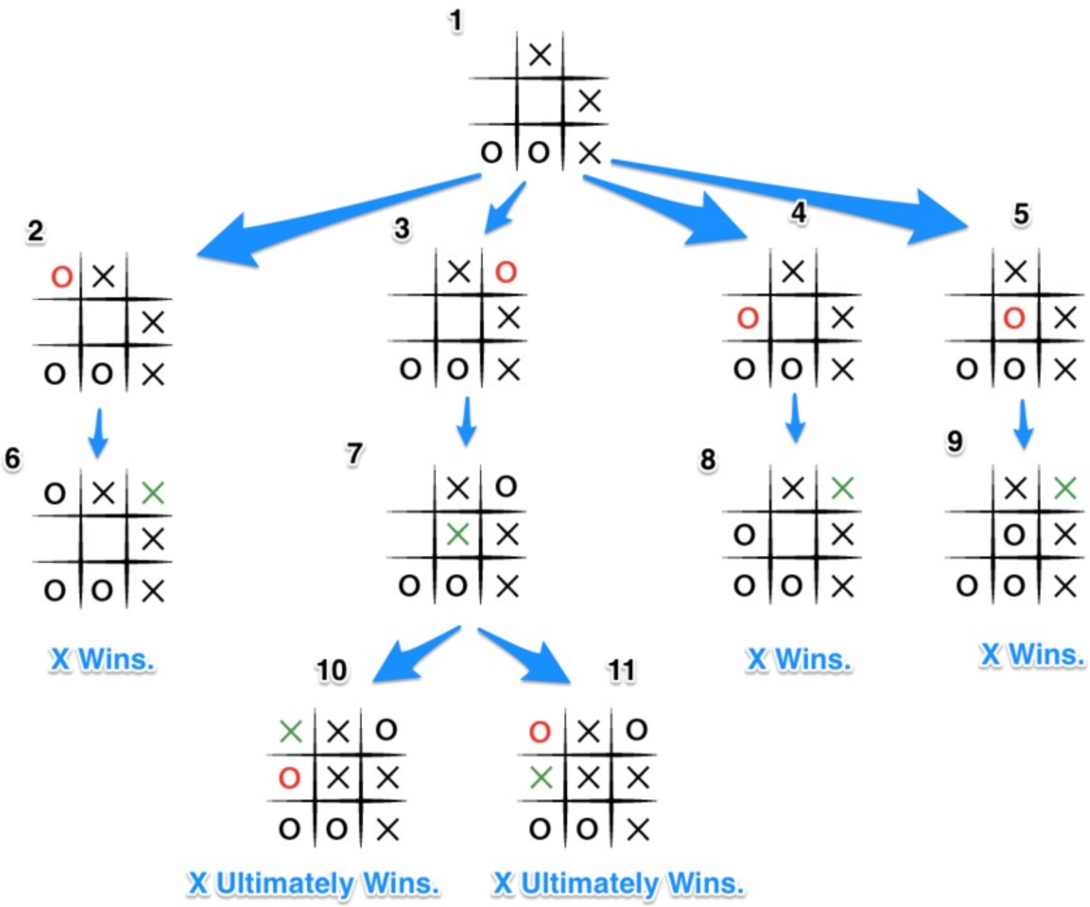
151 Retweets **14** Quote Tweets **569** Likes

Scene

Agents **search** and **plan**

using **heuristics** and **cost functions**

over **states.**



https://www.neverstopbuilding.com/blog/minimax

Uncertainty causes variability/variation

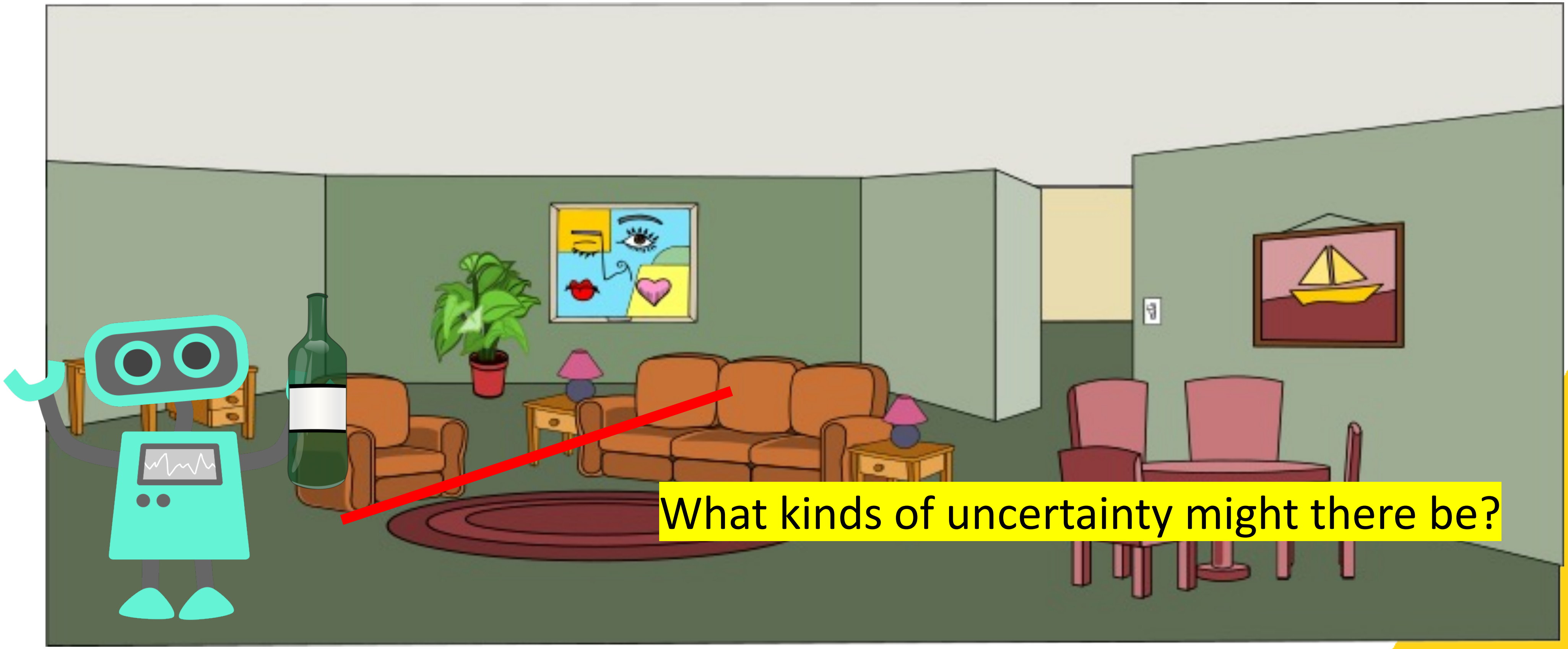
- Variation means branching
- Search: each edge is an action
  - Deterministic case: only actions and time change state
- Uncertainty: each action may land in a different state
- Example: one opponent token may be randomly deleted after your move
- How to model this?

Do stuff on board

Representing state

- Tic tac toe: can represent as a vector
- Uncertainty over values in the vector
    - Uncertainty due to game mechanics
- What about more complicated environments?

Example: Uncertainty in the Environment

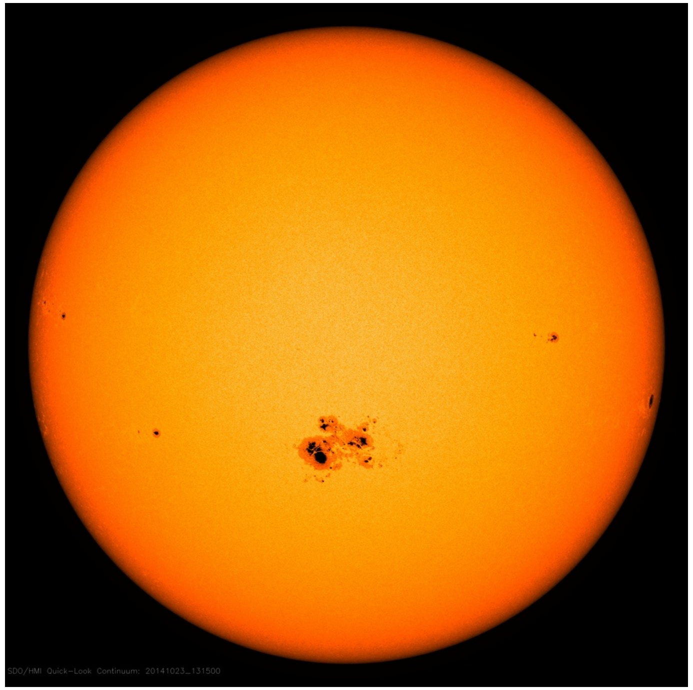


Sources of uncertainty

- **Object uncertainty:** presence or absence of an object
- **Measurement error:** bias that you might have uncertainty about
- **Low-frequency events**: may not have enough data to estimate
- **Environmental randomness**: truly random vs. population-based

Types of uncertainty

Epistemic (belief)



Aleatory ("truly random")

Implications for agent learning.

Break reasoning into events and objects

- States are composed of objects that have features that have values
- Can reason about values of those features with or without events
    - Without: probability of being in a certain state (not efficient!)
    - Events: relate actions and their effects
    - Actions may be external to the agent (e.g., performed by another agent)

Classic event example (sort of)

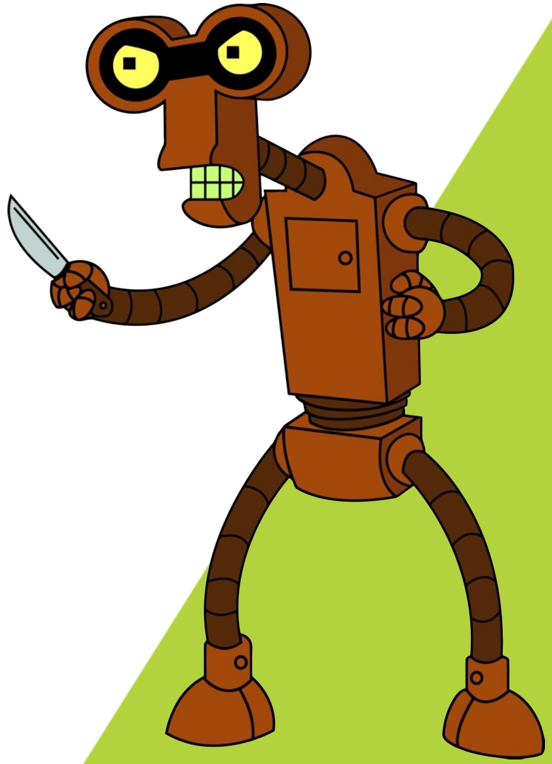
Scenario: Roberto is a defense robot with not currently hooked up to respond to your alarm system.

The alarm system responds to both burglaries and earthquakes.

You ask your neighbors Fry and Leela --- who are always home --- to switch him to "menace mode."

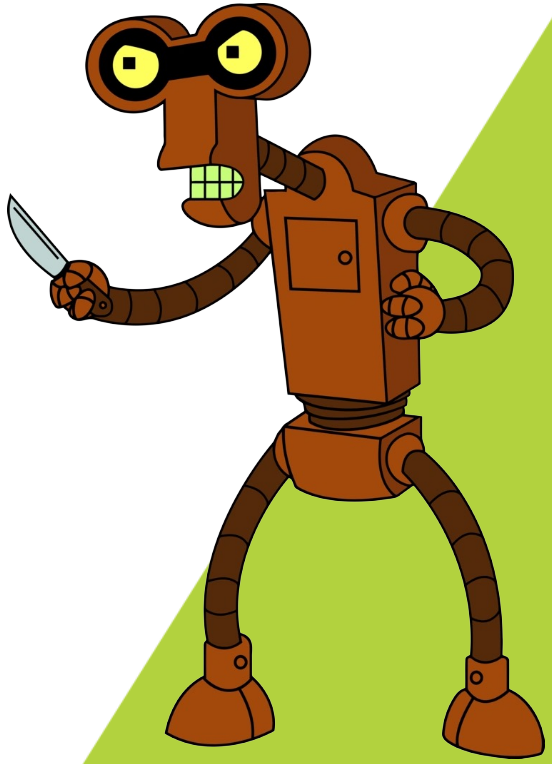
Leela always calls, but sometimes confuses the alarm with other noise.

Fry often misses the alarm, due to watching TV too loudly.



Where’s the state?

- Roberto has a representation of your house
- Roberto has a state for “acting menacing”
- Goal: decide (probabilistically) whether to “act menacing,” depending on who has called.



How to do this?

Suppose we have historic data in a table consisting of:

- Whether or not there was a burglary (B) of your house
- Whether or not there was an earthquake (E)
- Whether or not your alarm went off (A)
- Whether or not Fry called (F)
- Whether or not Leela called (L)

What do the following represent:

- One row in the table (R_i)?
- The count of duplicates of R_i (including R_i), divided by the total number of rows (p_j)?
- A table containing only unique R_i and their associated p_j?

Factorizing P(B, E, A, F, L)

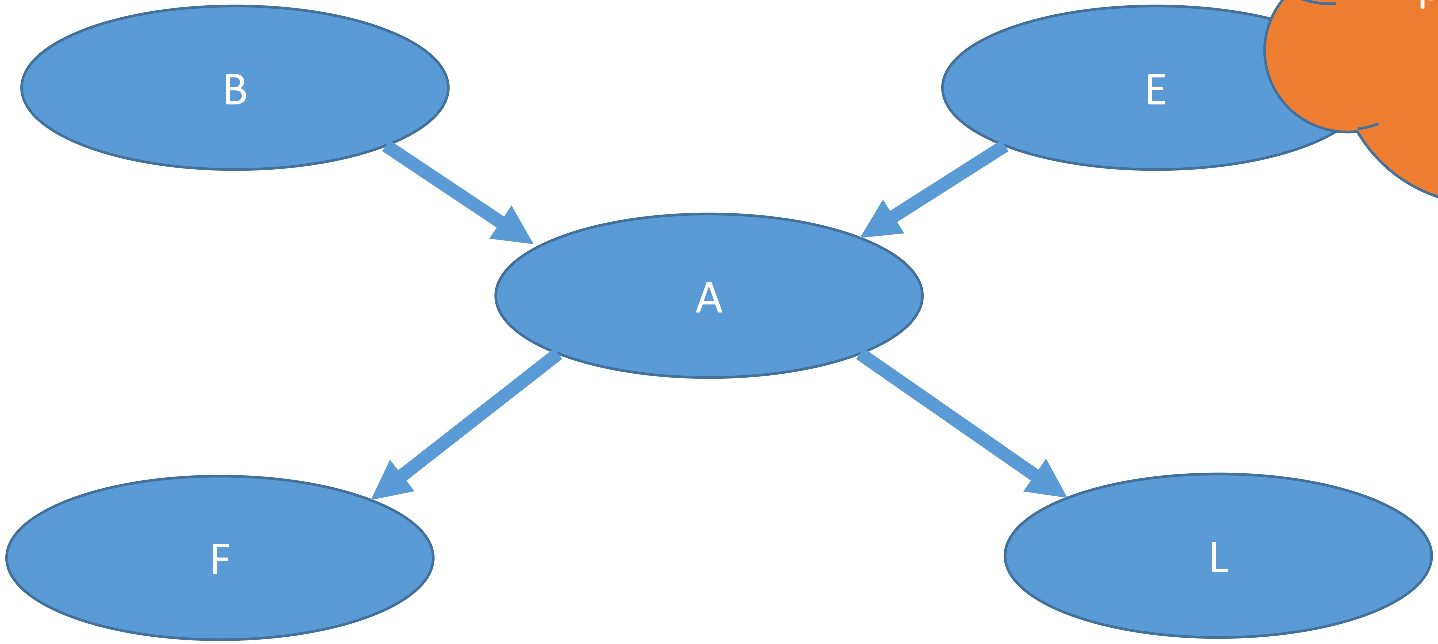
- Recall: $P(B, E, A, F, L) = P(B \mid E, A, F, L) * P(E \mid A, F, L) * P(A \mid F, L) * P(F \mid L) * P(L)$

  $= P(L \mid F, A, E, B) * P(F \mid A, E, B) * P(A \mid E, B) * P(E \mid B) * P(B)$

- Background knowledge: phone calls, alarms, and burglaries don't change whether an earthquake happens.
- Assumption: phone calls, alarms, and earthquakes don't change whether burglaries happen
- Only earthquakes and burglaries set off the alarm.
- Fry and Leela only sometimes activate Roberto

Factorizing P(B, E, A, F, L)

Idea: P(v | parents(v)) = P(v | parents (v) ∪ parents(parents(v))…)



$$P(B, E, A, F, L) = P(B \mid E, A, F, L) * P(E \mid A, F, L) * P(A \mid F, L) * P(F \mid L) * P(L)$$

$$= P(L \mid F, A, E, B) * P(F \mid A, E, B) * P(A \mid E, B) * \boxed{P(E \mid B)} * P(B)$$

P(L | A) P(F | A) P(E)

Do stuff on board

Practically…

Don’t need to rely on jointly collected data

Important for low-probability events!

- Can use historic data to better estimate P(E) and P(B)
  - Lower uncertainty over the parameters
- Can use manufacturer data for P(A | E, B)
- Then P(F | A) and P(L | A) are more about “belief”

Why factorize?

Do stuff on board

Usage

Bayes nets are a type of "probabilistic graphical model"

- When used for learning, popular across machine learning before deep learning
- Was especially popular in NLP
- Now: mostly known for causal reasoning (later topic)
    - Bleeding edge: renewed interest for explanation and fairness in deep learning

BLOG: Probabilistic Models with Unknown Objects*

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong and **Andrey Kolobov**
Computer Science Division
University of California
Berkeley, CA 94720-1776
{milch, bhaskara, russell, dsontag, dlong, karaya1}@cs.berkeley.edu

Abstract

This paper introduces and illustrates BLOG, a formal language for defining probability models over worlds with unknown objects and identity uncertainty. BLOG unifies and extends several existing approaches. Subject to certain acyclicity constraints, every BLOG model specifies a unique probability distribution over first-order model structures that can contain varying and unbounded numbers of objects. Furthermore, complete inference algorithms exist for a large fragment of the language. We also introduce a probabilistic form of Skolemization for handling evidence.

1 Introduction

Human beings and AI systems must convert sensory input into some understanding of what's out there and what's going on in the world. That is, they must make inferences about the objects and events that underlie their observations. No pre-specified list of objects is given; the agent must infer the existence of objects that were not known initially to exist.

In many AI systems, this problem of unknown objects is engineered away or resolved in a preprocessing step. However, there are important applications where the problem is unavoidable. *Population estimation*, for example, involves counting a population by sampling from it randomly and measuring how often the same object is resampled; this would be pointless if the set of objects were known in advance. *Record linkage*, a task undertaken by an industry of more than 300 companies, involves matching entries across multiple databases. These companies exist because of uncertainty about the mapping from observations to underlying objects. Finally, *multi-target tracking* systems perform *data association*, connecting, say, radar blips to hypothesized aircraft.

Probability models for such tasks are not new: Bayesian models for data association have been used since the 1960s [Sittler, 1964]. The models are written in English and mathematical notation and converted by hand into special-purpose code. In recent years, *formal representation languages* such as graphical models [Pearl, 1988] have led to general inference algorithms, more sophisticated models, and automated model selection (structure learning). In Sec. 7, we review several *first-order probabilistic languages* (FOPLs) that explicitly represent objects and the relations between them. However, most FOPLs only deal with fixed sets of objects, or deal with unknown objects in limited and *ad hoc* ways. This paper introduces BLOG (Bayesian LOGic), a compact and intuitive language for defining probability distributions over outcomes with varying sets of objects.

We begin in Sec. 2 with three example problems, each of which involves possible worlds with varying object sets and identity uncertainty. We describe generative processes that produce such worlds, and give the corresponding BLOG models. Sec. 3 observes that these possible worlds are naturally viewed as model structures of *first-order logic*. It then defines precisely the set of possible worlds corresponding to a BLOG model. The key idea is a generative process that constructs a world by adding objects whose existence and properties depend on those of objects already created. In such a process, the existence of objects may be governed by many random variables, not just a single population size variable. Sec. 4 discusses how a BLOG model specifies a probability distribution over possible worlds.

Sec. 5 solves a previously unnoticed "probabilistic Skolemization" problem: how to specify evidence about objects—such as radar blips—that one didn't know existed. Finally, Sec. 6 briefly discusses inference in unbounded outcome spaces, stating a sampling algorithm and a completeness theorem for a large class of BLOG models, and giving experimental results on one particular model.

2 Examples

In this section we examine three typical scenarios with unknown objects—simplified versions of the population estimation, record linkage, and multitarget tracking problems mentioned above. In each case, we provide a short BLOG model that, when combined with a suitable inference engine, constitutes a working solution for the problem in question.

Example 1. *An urn contains an unknown number of balls—say, a number chosen from a Poisson distribution. Balls are equally likely to be blue or green. We draw some balls from the urn, observing the color of each and replacing it. We cannot tell two identically colored balls apart; furthermore, observed colors are wrong with probability 0.2. How many balls are in the urn? Was the same ball drawn twice?*

The BLOG model for this problem, shown in Fig. 1, describes a stochastic process for generating worlds. The first 4

*This work was supported by DARPA under award 03-000219, and by an NSF Graduate Research Fellowship to B. Milch.

State uncertainty: foundational & active

- Work published 2005-2009
- Milch's dissertation: very thorough

Kozen semantics -- theoretical CS
(FOCS 1979)

Church -- artificial intelligence
(UAI 2008)

Scenic – programming languages
(PLDI 2019)

Next class: hands-on exercises using Bayes nets
Introduce partial observability
(bring your laptops)